

**BTS SIO – Documentation mission E5**

|  |  |  |  |
| --- | --- | --- | --- |
| Rédacteur(s) | Version | Date | Nb pages |
| Zatric Nihad | 1.1 | 04/04/2021 | 18 |

**AUTHENTIFICATION**

**SOMMAIRE**

[1 Explication du document au jury 3](#_Toc100046917)

[2 Démarrage du projet 5](#_Toc100046918)

[2.1 Charte de projet 5](#_Toc100046919)

[2.2 Expression fonctionnelle du besoin 6](#_Toc100046920)

[2.3 Contraintes 7](#_Toc100046921)

[2.4 Gestion des droits d’accès 7](#_Toc100046922)

[3 Conception de l’application 8](#_Toc100046923)

[3.1 Schématisation de l'application 8](#_Toc100046924)

[3.2 Choix des technologies employées 9](#_Toc100046925)

[3.3 Présentation des composants logiciels 9](#_Toc100046926)

[4 Conception de la BDD 10](#_Toc100046927)

[4.1 Schéma de la base de données 10](#_Toc100046928)

[4.2 Installation et configuration de la base de données 10](#_Toc100046929)

[4.3 Interaction de l'application avec la base de données 11](#_Toc100046930)

[5 Gestion de la qualité 12](#_Toc100046931)

[5.1 Méthodologie et outil de versioning 12](#_Toc100046932)

[5.2 Gestion des tests de l'application 13](#_Toc100046933)

[5.3 Rédaction de la documentation 16](#_Toc100046934)

[6 Mise en production 17](#_Toc100046935)

[6.1 Description de l'environnement de développement 17](#_Toc100046936)

[6.2 Description de l'environnement de production 17](#_Toc100046937)

[6.3 Etape de mise en production 18](#_Toc100046938)

[6.4 Mise en place de la sauvegarde 18](#_Toc100046939)

[7 Bilan du projet 18](#_Toc100046940)

# Explication du document au jury

Le plan de ce document se base sur un plan type conçu par notre pilote de formation afin de nous aider à travailler en mode projet durant la réalisation de nos missions E5. J’ai simplifié ce plan pour ne garder que les parties traitées dans ma mission.

**Explication du plan type :**

Le plan type se base sur la circulaire du BTS SIO disponible sur le site du réseau certa (<https://www.reseaucerta.org/sio/circulaire>). Les compétences sont numérotées en fonction des 3 blocs :

1. Concevoir et développer une solution applicative
2. Assurer la maintenance corrective ou évolutive d’une solution applicative
3. Gérer les données

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Circulaire BTS SIO 2022 | |
| Plan du document | | Compétences de bloc correspondantes | Indicateurs de performance correspondants |
| Démarrage du projet | | | |
|  | Charte de projet | 1.1 Analyser un besoin exprimé et son contexte juridique | La proposition de la solution applicative répond au besoin exprimé dans le cahier des charges |
| Expression fonctionnelle du besoin | - la maquette des éléments applicatifs de la solution respecte les fonctionnalités exprimées ; |
| Contraintes | L’interface utilisateur est mise à jour en respectant les contraintes ergonomiques. |
| Gestion des droits d’accès | Les accès aux données sont contrôlés conformément aux habilitations définies par le cahier des charges. |
| Conception de l'application | | | |
|  | Schématisation de l'application | 1.2 Participer à la conception de l’architecture d’une solution applicative |  |
| 1.3 Modéliser une solution applicative |  |
| Choix des technologies employées | 1.4 Exploiter les ressources du cadre applicatif (framework) |  |
| Présentation des composants logiciels | 1.5 Identifier, développer, utiliser ou adapter des composants logiciels | Le choix des composants logiciels à utiliser et/ou à développer est pertinent. |
| le développement tient compte des préoccupations de développement durable. |
| Des composants logiciels sont adaptés pour améliorer la qualité de la solution applicative. |
| 1.6 Exploiter les technologies Web pour mettre en œuvre les échanges entre applications, y compris de mobilité | Un service Web est exploité pour échanger des données entre applications. |
| Conception de la base de données | | | |
|  | Schéma de la base de données | 3.3 Concevoir ou adapter une base de données | Les données sont modélisées conformément au besoin de la solution applicative. |
| Installation et configuration de la base de données | 3.4 Administrer et déployer une base de données |  |
| Interaction de l'application avec la base de données | 1.7 Utiliser des composants d’accès aux données | Les données persistantes liées à la solution applicative sont exploitées à travers un langage de requête lié à la base de données qui peut être le langage de requête proposé par les échanges applicatifs des technologies Web, un langage de requête présent dans l’outil de correspondance objet-relationnel ou toute autre solution de persistance. |
| 3.1 Exploiter des données à l’aide d’un langage de requêtes |  |
| Gestion de la qualité | | | |
|  | Méthodologie et outil de versioning | 1.8 Intégrer en continu les versions d’une solution applicative | Un outil collaboratif de gestion des versions est utilisé. |
| Gestion des tests de l'application |  | Les tests de non régression sont réalisés. |
| Gestion des tests de l'application | 1.9 Réaliser les tests nécessaires à la validation ou à la mise en production d’éléments adaptés ou développés | Les composants logiciels sont validés par les procédures de tests unitaires et fonctionnels. |
| un document est rédigé pour chaque contexte d’utilisation de l’application et est adapté à chaque destinataire tant par son contenu que par sa présentation ; |
| 1.11 Exploiter les fonctionnalités d’un environnement de développement et de tests |  |
| Rédaction de la documentation | 1.10 Rédiger des documentations technique et d’utilisation d’une solution applicative |  |
| Mise en production | | | |
|  | Description de l'environnement de développement |  | Le choix du type de base de données est pertinent. |
| Description de l'environnement de production |  |  |
| Etapes de mise en production |  | La base de données est opérationnelle et stable dans l’environnement de production. |
|  | L’application développée est opérationnelle conformément au cahier des charges et stable dans l’environnement de production. |
| Mise en place de la sauvegarde |  | Des tests de restauration sont effectués. |
|  | La base de données est sauvegardée selon la planification retenue. |
| Gestion de la maintenance (corrective / évolutive) | | | |
|  | Mise à jour de la documentation du SI | 2.1 Recueillir, analyser et mettre à jour les informations sur une version d’une solution applicative | Les composants logiciels sont documentés de manière à être réutilisés. |
| La documentation technique et d’utilisateurs de la solution applicative sont mises à jour. |
| 2.4 Mettre à jour des documentations technique et d’utilisation d’une solution applicative | - une documentation des versions vient appuyer l’intégration continue ; |
| - les composants logiciels sont documentés de manière à être réutilisés ; |
| Evaluation de la qualité du livrable | 2.2 Évaluer la qualité d’une solution applicative |  |
| Procédure de correction d'un dysfonctionnement | 2.3 Analyser et corriger un dysfonctionnement | Le dysfonctionnement de la solution existante est corrigé selon les procédures en vigueur et dans les délais. |
| Gestion des tests de mise à jour | 2.5 Élaborer et réaliser les tests des éléments mis à jour | Les composants logiciels adaptés et/ou corrigés sont validés par les procédures de tests unitaires et fonctionnels. |
| Bilan du projet | | | |
|  | Bilan |  |  |

# Démarrage du projet

## Charte de projet

|  |
| --- |
| **Type de mission** |
| Mission effectuée dans le cadre d’un projet personnel. |
| **Contexte** |
| Dans le cadre d’un projet personnel, j’ai réalisé un modèle de formulaire de connexion à la base de données. |
| **Demande du client** |
| Réaliser un formulaire permettant à un administrateur de se connecter via un nom de compte et un mot de passe. |
| **Budget disponible** |
| Matériel : Un ordinateur avec un serveur Wamp, un hébergement o2switch.  Licences : PhpStorm |
| **Outils disponibles** |
| PhpStorm |
| **Confidentialité** |
| Libre. |

## Expression fonctionnelle du besoin

Liste des fonctionnalités attendues :

|  |
| --- |
| **Front office** |
| Page de destinations :  Le formulaire est sur la page de destinations. Il est caché, et apparaît lorsqu’on clique sur le bouton « connexion » en haut à droite de la page. |

|  |
| --- |
| **Back Office** |
| Le back office est constitué d’un fichier PHP : sql\_authentification.php.  Ce fichier permet d’effectuer une requête MySQL afin de récupérer tous les comptes administrateurs dans la base de données. Si il y a une correspondance, on démarre une session en remplissant des variables de session avec les informations de l’administrateur, pour lui permettre d’accéder à des fonctionnalités avancées du programme. |

## Contraintes

|  |
| --- |
| **Générales** |
| Temps : 2 jours. |
| **Juridiques** |
| Aucune donnée utilisateur ne sera traitée ou récoltée par le site. |
| **Techniques** |
| Requête MySQL  Hachage de mot de passe (password\_hash et password\_verify).  Evènements javascript. |
| **Ergonomique** |
| Responsive. |

## Gestion des droits d’accès

|  |
| --- |
| **Administrateur** |
| Moi-même. |

|  |
| --- |
| **Visiteurs** |
| Toute personne possédant l’URL de mon site. |

# Conception de l’application

## Schématisation de l'application

Une image contenant table

Description générée automatiquement

## Choix des technologies employées

Langage serveur PHP couplé au langage de données MySQL. Grâce à ces deux langages, je peux interagir avec la base de données afin de vérifier le nom de compte et le mot de passe entré par l’administrateur.

Langage Javascript pour le contrôle du formulaire, et pour les différents évènements.

Pas besoin de framework puisque le but est de réaliser un système d’authentification sans framework, de sorte à pouvoir le réutiliser ultérieurement dans d’autres projets.

Je n’ai pas jugé utile d’employer la méthode AJAX même si cela aurait pu être pertinent.

## Présentation des composants logiciels

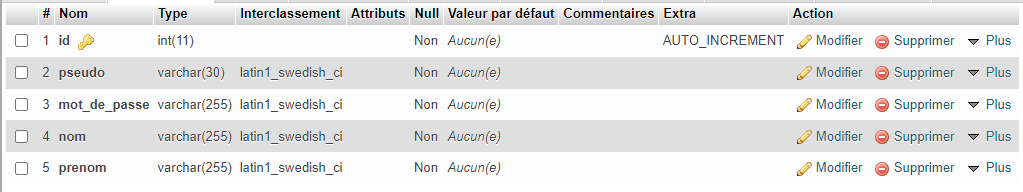
<https://www.php.net/manual/fr/function.password-hash.php>

https://www.php.net/manual/fr/reserved.variables.session.php

# Conception de la BDD

## Schéma de la base de données

Les données sont modélisées conformément au besoin de la solution applicative.



5 champs :

* Id : auto-incrémenté, il permet de différencier les administrateurs présents dans la BDD.
* Pseudo : nom de compte de l’utilisateur.
* Mot\_de\_passe : mot de passe de l’utilisateur.
* Nom : nom de l’utilisateur.
* Prénom : prénom de l’utilisateur.

## Installation et configuration de la base de données

L’outil de base de données utilisé est PHP my admin pour le langage MySQL.

Les deux sont automatiquement installés en local grâce au serveur Wamp, et sur le site en ligne sur Cpanel.

## Interaction de l'application avec la base de données

**Requête de récupération des comptes administrateurs stockés dans la base de données :**

$query = "SELECT \* FROM utilisateurs";

Cette requête permet de sélectionner tous les éléments dans la table utilisateurs.

# Gestion de la qualité

## Méthodologie et outil de versioning

Pour intégrer en continu les versions de la solution applicative, j’utilise l’outil de versioning Github.

Puisque je suis le seul à l’utiliser, je n’ai pas besoin d’autres branches que la branche Main. Au fur et à mesure des modifications de la solution applicative, je « pousse » les modifications vers Github, en laissant un commentaire afin d’identifier la modification réalisée. Cela me permet d’avoir une trace des modifications que j’effectue.

Une image contenant texte, moniteur, capture d’écran, intérieur

Description générée automatiquement

## Gestion des tests de l'application

**Unitaires :**

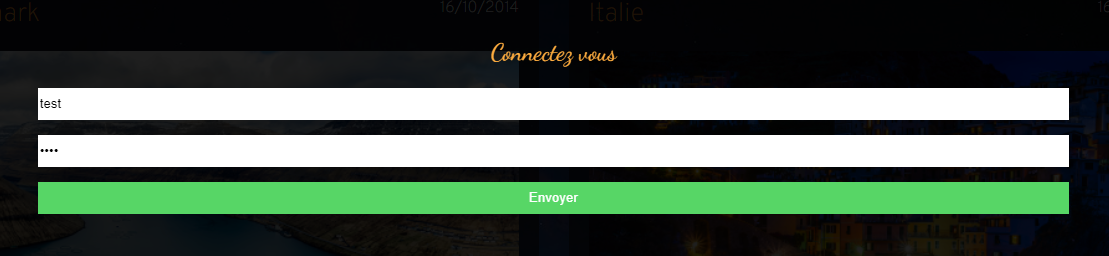
Grâce à des var\_dump en PHP et à des console.log en javascript (fonctions permettant de renvoyer des variables), je teste points par points le logiciel tout en le créant.

**Fonctionnels :**

Je me mets à la place de l’utilisateur final et je teste deux cas de figure :

**Premier contexte, mauvais identifiants :**

J’essaye de me connecter avec des identifiants erronés :



La page est actualisée mais la connexion est refusée, le formulaire est de nouveau affiché via une fonction javascript et le message d’erreur apparaît :

Une image contenant texte

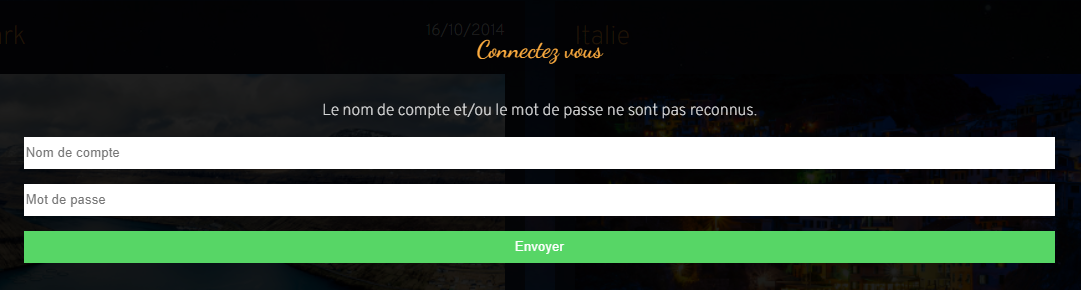
Description générée automatiquement

Fichier echec\_authentification.php :

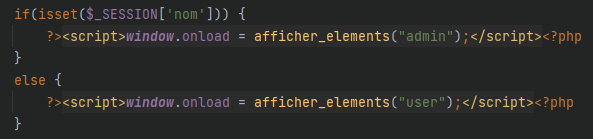
Une image contenant texte

Description générée automatiquement

Ce qui renvoie d’un point de vue utilisateur :



De plus, la fonction afficher\_elements est lancée avec le paramètre ‘user’ puisqu’il n’y a pas de session créée : cette fonction va chercher de façon asynchrone toutes les destinations dans la base de données pour créer la page.

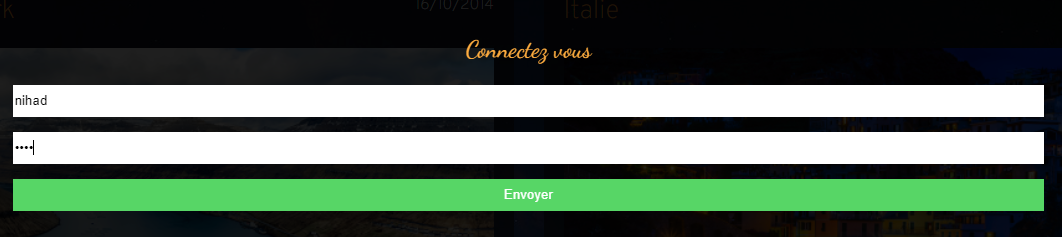


Le visuel des destinations est le suivant :

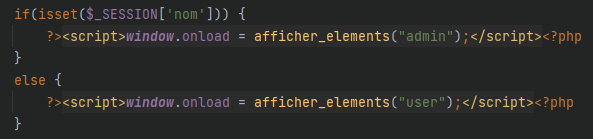
Une image contenant texte, équipement électronique, moniteur, afficher

Description générée automatiquement

Contexte 2 : Je me connecte avec les bons identifiants :



La page est actualisée, et la fonction afficher\_elements qui permet d’afficher toutes les destinations via une requête asynchrone est lancée avec le paramètre ‘admin’, ce qui permet à la fonction d’afficher des éléments de contrôle supplémentaires (boutons de modification, de suppression etc…) :



Le visuel des destinations est le suivant :

Une image contenant texte, équipement électronique, moniteur, afficher

Description générée automatiquement

Un bouton de création de destination apparaît :

Une image contenant texte

Description générée automatiquement

Et le bouton de connexion est remplacé par le nom et le prénom de l’administrateur :

Une image contenant texte

Description générée automatiquement

Lorsqu’on clique sur ce nom, l’administrateur est déconnecté.

## Rédaction de la documentation

Puisque je suis le seul à posséder un compte administrateur, il n’est pas nécessaire de rédiger une documentation technique. Cependant, si à l’avenir une autre personne venait à être amenée à utiliser le logiciel, il faudrait en rédiger une.

# Mise en production

## Description de l'environnement de développement

Windows 10 comme système d’exploitation.

PhpStorm comme IDE.

Wampserver comme serveur local.

MySQL couplé à PhpMyAdmin comme SGBD.

Je suis à l’aise avec cet environnement de travail car c’est le même dans l’entreprise ou je suis apprenti.

J’ai choisi o2switch comme hébergement à cause des bons avis concernant le service client.

## Description de l'environnement de production

L’environnement de production est un serveur mutualisé hébergé chez o2switch.



## Etape de mise en production

Une fois que toutes les fonctionnalités essentielles de l’application ont été réalisées et testées et que le cahier des charges a été respecté, j’ai mis en production la page de destinations.

## Mise en place de la sauvegarde

Puisque cette application n’a aucun but autre que de me fournir un modèle pour des travaux personnels ultérieurs, aucune sauvegarde de la base de données n’est nécessaire.

Cependant, il aurait pu être réalisé grâce à o2switch.

# Bilan du projet

Ce projet m’a permis entre autres de me familiariser avec les évènements JS, mais aussi et surtout avec le hachage de mots de passes. De plus, je comprends mieux le fonctionnement des systèmes d’authentification.

De ce fait, je possède une base solide réutilisable lors de projets ultérieurs.